MATH 3027 Optimization 2021: Coursework 2

Mingdong He[*]

Question 1

(a): rewrite this problem in matrix form:

$$f = x_1^2 + 2x_2^2 - 3x_3 = \mathbf{x^T} A\mathbf{x} + b^T\mathbf{x}$$

where $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, $b^T = \begin{pmatrix} 0 & 0 & 3 \end{pmatrix}$ the function is convex since the matrix A is diagonally dominant.

The constraint is a unit simplex Δ_3, which is convex.

However, this is not a convex problem as we are maximizing a convex function.

Instead, f is non-constant, continuous, convex over Δ_3, there exists one maximizer, which is an extreme point of Δ_3. Extreme points are $(1,0,0)^T, (0,1,0)^T, (0,0,1)^T$. $f((1,0,0)^T) = 1, f((0,1,0)^T) = 2, f((0,0,1)^T) = -3$, so optimal point is $(0,1,0)$ and $f^* = 2$.

(b): Let's rewrite the function as matrix form: $x_1^2 + 2x_2^2 - 3x_3 = \mathbf{x^T} A\mathbf{x}$, where $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$, so that the matrix is indefinite considering the eigenvalues. This problem is not convex.

Let's try to write the dual problem:

Lagrangian:

$$L = x_1^2 - x_2^2 - x_3^2 + \lambda(x_1^4 + x_2^4 + x_3^4 - 1)$$

$$\nabla_x L = \begin{pmatrix} 2x_1 + 4\lambda x_1^3 \\ -2x_2 + 4\lambda x_2^3 \\ -2x - 3 + 4\lambda x_3^3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Solve this equations, since $\lambda \geq 0$, we have:

$$x_1(2 + 4\lambda x_1^2) = 0 \implies x_1 = 0$$

$$x_2(-2 + 4\lambda x_2^2) = 0 \implies x_2 = 0 \; or \; x_2^2 = \frac{1}{2\lambda}$$

$$x_3(-2 + 4\lambda x_3^2) = 0 \implies x_3 = 0 \; or \; x_3^2 = \frac{1}{2\lambda}$$

Note that here we keep the square due to our original function, i.e, the lowest power for terms including x_1, x_2, x_3 are 2. So potential values to minize L are: $(0,0,0), (0, \frac{1}{2\lambda}, \frac{1}{2\lambda}), (0, 0, \frac{1}{2\lambda}), (0, \frac{1}{2\lambda}, 0)$ Substitute these

[*]School of Mathematical Sciences, University of Nottingham, smymh1@nottingham.ac.uk

values into lagrangian:

$$L\left((0,0,0)\right) = -\lambda$$
$$L\left((0, \frac{1}{2\lambda}, \frac{1}{2\lambda})\right) = -\frac{1}{2\lambda} - \lambda$$
$$L\left((0,0,\frac{1}{2\lambda})\right) = -\frac{1}{4\lambda} - \lambda$$
$$L\left((0,\frac{1}{2\lambda},0)\right) = -\frac{1}{4\lambda} - \lambda$$

Obviously, since $\lambda \geq 0$, $L\left((0, \frac{1}{2\lambda}, \frac{1}{2\lambda})\right)$ is the minima. Thus, $\max_{\lambda \in \mathbb{R}}\ q(\lambda) = \max_{\lambda \in \mathbb{R}}\ L\left((0, \frac{1}{2\lambda}, \frac{1}{2\lambda})\right) = -\sqrt{2}$, since the dual objective function $q(\lambda) = L\left((0, \frac{1}{2\lambda}, \frac{1}{2\lambda})\right) = -\frac{1}{2\lambda} - \lambda \leq -\sqrt{2}$, when $\lambda = \frac{\sqrt{2}}{2}$ the equality holds. Therefore, when $(x_1^2, x_2^2, x_3^2) = (0, \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$. i.e: $(x_1, x_2, x_3) = \left(0, (\frac{1}{2})^{\frac{1}{4}}, (\frac{1}{2})^{\frac{1}{4}}\right)$, or $\left(0, (\frac{1}{2})^{\frac{1}{4}}, -(\frac{1}{2})^{\frac{1}{4}}\right)$, or $\left(0, -(\frac{1}{2})^{\frac{1}{4}}, (\frac{1}{2})^{\frac{1}{4}}\right)$, or $\left(0, -(\frac{1}{2})^{\frac{1}{4}}, -(\frac{1}{2})^{\frac{1}{4}}\right)$, we could get the minimal value $-\sqrt{2}$ for primal problem since the slater's condition satisfied (i.e: choose $(0,0,0)$, the inequality constraint strickly holds).

Question 2

1. This is a convex optimization problem. The objective function is linear, which is a convex function and the constraints is a convex set since it is a circle encompassing all the points. Therefore, it is a convex problem.

2.

$$
\begin{aligned}
&Lagrangian:\\
&L = \gamma + \sum_{i=1}^{m} \lambda_i(||\mathbf{x} - \mathbf{a_i}||_2^2 - \gamma), i = 1, ..., m\\
\\
&KKT:\\
&\nabla_x \gamma + \nabla_x \sum_{i=1}^{m} (\lambda_i(||\mathbf{x} - \mathbf{a_i}||_2^2 - \gamma)) = 0, i = 1, ..., m.\\
&\implies 2\sum_{i=1}^{m} \lambda_i(\mathbf{x} - \mathbf{a_i}) = 0\\
&Primal\ feasibility:\\
&||\mathbf{x} - \mathbf{a_i}||_2^2 - \gamma \leq 0, i = 1, ..., m\\
\\
&Dual\ feasibility:\\
&\lambda_i \geq 0, i = 1, ..., m\\
\\
&Complementary\ Slackness:\\
&\lambda_i(||\mathbf{x}^* - \mathbf{a_i}||_2^2 - \gamma) = 0, i = 1, ..., m
\end{aligned}
$$

3. Let's first write Lagrangian:

$$\begin{aligned}
&Lagrangian:\\
&L = \gamma + \sum_{i=1}^{m} \lambda_i(||\mathbf{x} - \mathbf{a_i}||_2^2 - \gamma), i = 1, ..., m
\end{aligned}$$

Since we need to minimize the above equation, so:

$$\begin{aligned}
\nabla_x L &= 2\sum_{i=1}^{m} \lambda_i(\mathbf{x} - \mathbf{a_i}) = 0\\
\implies \mathbf{x}^* &= \frac{\sum_{i=1}^{m} \lambda_i \mathbf{a_i}}{\sum_{i=1}^{m} \lambda_i}
\end{aligned}$$

Expand L: substitute the above x^*. Since we know that $\gamma \geq 0, \lambda_i \geq 0$:

$$\begin{aligned}
q(\lambda) = \min_{x\in\mathbb{R}^2}\ L &= \min_{x\in\mathbb{R}^2} \left\{ \gamma + \sum_{i=1}^{m} \lambda_i(||\mathbf{x} - \mathbf{a_i}||_2^2 - \gamma) \right\}\\
&= \min_{x\in\mathbb{R}^2} \left\{ \sum_{i=1}^{m} \lambda_i(\mathbf{x} - \mathbf{a_i})^T(\mathbf{x} - \mathbf{a_i}) + \gamma(1 - \sum_{i=1}^{m} \lambda_i) \right\}\\
&= \min_{x\in\mathbb{R}^2} \left\{ \sum_{i=1}^{m} \lambda_i(||\mathbf{x}||^2 - 2\mathbf{x}^T\mathbf{a_i} + ||\mathbf{a_i}||^2) + \gamma(1 - \sum_{i=1}^{m} \lambda_i) \right\}\\
&= \min_{x\in\mathbb{R}^2} \left\{ ||\mathbf{x}||^2 \sum_{i=1}^{m} \lambda_i - 2\mathbf{x}^T \sum_{i=1}^{m} \lambda_i \mathbf{a_i} + \sum_{i=1}^{m} \lambda_i ||\mathbf{a_i}||^2 + \gamma(1 - \sum_{i=1}^{m} \lambda_i) \right\}, substitute\ x^*\\
&= \frac{(\sum_{i=1}^{m} \lambda_i \mathbf{a_i})^2}{\sum_{i=1}^{m} \lambda_i} - 2\frac{(\sum_{i=1}^{m} \lambda_i \mathbf{a_i})^2}{\sum_{i=1}^{m} \lambda_i} + \sum_{i=1}^{m} \lambda_i ||\mathbf{a_i}||^2 + \gamma(1 - \sum_{i=1}^{m} \lambda_i)\\
&= \underbrace{-\frac{(\sum_{i=1}^{m} \lambda_i \mathbf{a_i})^2}{\sum_{i=1}^{m} \lambda_i}}_{\leq 0} + \underbrace{\sum_{i=1}^{m} \lambda_i ||\mathbf{a_i}||^2}_{\geq 0} + \underbrace{\gamma(1 - \sum_{i=1}^{m} \lambda_i)}_{\leq 0}\\
&= -(\sum_{i=1}^{m} \lambda_i \mathbf{a_i})^2 + \sum_{i=1}^{m} \lambda_i ||\mathbf{a_i}||^2, if\ \sum_{i=1}^{m} \lambda_i = 1.\ or\ -\infty, otherwise.
\end{aligned}$$

Let's rewrite this result into matrix form and the dual problem requires us to maximize the dual objective function $q(\lambda)$, thus the dual problem is:

$$\begin{aligned}
&\max_{\lambda}\ -||\mathbf{A}\boldsymbol{\lambda}||^2 + \sum \lambda_i ||a_i||^2\\
&s.t. \sum_{i=1}^{m} \lambda_i = 1\\
&\qquad \lambda_i \geq 0, i = 1, ...m
\end{aligned}$$

where:$\mathbf{A} = (a_1, a_2, .., a_m) \in \mathbb{R}^{2\times m}$, since $a_1, a_2, ..., a_m \in \mathbb{R}^2$, and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_m)^T$.

4. **Pseudocode Algorithm**:

Initialization: A tolerance parameter ϵ and $\boldsymbol{\lambda}^0 \in \mathbb{R}^m$. $f = -||\mathbf{A}\lambda||^2 + \sum_{i=1}^m \lambda_i ||\mathbf{a_i}||^2 = -\lambda^\mathbf{T}\mathbf{A}^\mathbf{T}\mathbf{A}\lambda + \mathbf{b}^T\lambda$, $\nabla f = -2\mathbf{A}^\mathbf{T}\mathbf{A}\lambda + \mathbf{b}$, where $\mathbf{b} = \left(||\mathbf{a_1}||^2, ..., ||\mathbf{a_m}||^2\right)^T$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2.., \lambda_m)^T$. (Both $\mathbf{b}$ and $\boldsymbol{\lambda}$ are column vectors).

General Step: for $k = 0, 1, 2...$ execute the following steps:

1. Pick $t^k = \frac{1}{L}$, L is Lipschitz constant of the gradient of the objective function, where $L = 2||\mathbf{A}^\mathbf{T}\mathbf{A}||_2$.

2. $\boldsymbol{\lambda}_{k+1} = [\boldsymbol{\lambda}_k + t^k \nabla f(\boldsymbol{\lambda}_k)]_+$.

3. If $||\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1}|| \le \epsilon$, then STOP. The optimal is $\mathbf{x}^* = \mathbf{A}\boldsymbol{\lambda}^{k+1}$.

5. The Lagrangian is:

$$\begin{aligned}
L(\mathbf{x}, \mu) &= ||\mathbf{x} - \mathbf{y}||^2 + 2\mu(\mathbf{e}^T\mathbf{x} - 1), for\ \mathbf{x} \ge 0. \\
&= ||\mathbf{x}||^2 - 2(\mathbf{y} - \mu\mathbf{e})^T\mathbf{x} + ||\mathbf{y}||^2 - 2\mu \\
&= \sum_{i=1}^m (x_i^2 - 2(y_i - \mu)x_i) + ||\mathbf{y}||^2 - 2\mu
\end{aligned}$$

Take the gradient of Lagrangian, we get:

$$\begin{aligned}
\nabla_x L(\mathbf{x}, \mu) &= \nabla_x (\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y}) + 2\mathbf{e}\mu \\
&= 2(\mathbf{x} - \mathbf{y}) + 2\mathbf{e}\mu = 0
\end{aligned}$$

This problem is separable with respect to the variables x_i and hence the optimal x_i is the solution to the one-dimensional problem:

$$\min_{x_i \ge 0} [x_i^2 - 2(y_i - \mu)x_i]$$

Let's write it as: $f(x) = x_i^2 - 2(y_i - \mu)x_i$, and $f'(x) = 2x_i - 2(y_i - \mu) = 0$, we can get:$x_i = y_i - \mu$. However, since $\mathbf{x} \ge 0$, there are two cases:

1. if $y_i - \mu \ge 0$, the function could reach the minimum since this is a convex function with the turning point on the positive x-axis, which is indeed $x_i = y_i - \mu$.

2. if $y_i - \mu \le 0$, the turning point of the function is on the negative x-axis, so the function is increasing for $x_i \ge 0$, so the minmum could be reached at $x_i = 0$.

Thus, the optimal solution to the above problem is given by: $x_i = y_i - \mu$, if $y_i \ge \mu$, 0, otherwise, which is

$$x_i = [y_i - \mu]_+$$

The optimal value is $-[y_i - \mu]_+^2$. Therefore, the dual problem is:

$$\max_\mu g(\mu) = -\sum_{i=1}^m [y_i - \mu]_+^2 - 2\mu + ||\mathbf{y}||^2$$

with the minimum achieved at $x_i^* = [y_i - \mu]_+$.

Bisection method:

```
bisection <- function(f,lb,ub,eps){
  stopifnot(f(lb)*f(ub)<0, lb<ub)
  iter=0
  while (ub-lb>eps){
    z=(lb+ub)/2
    iter=iter+1
    if(f(lb)*f(z)>0){
      lb=z
      }
    else{
      ub=z
    }
  }
  return(z)
}
```

Code for the function to solve $g'(\mu)$ by bisection method:

```
proj<-function(x){
  pmax(x,0)
}

#projection onto a simplex
projsimplex<-function(y){
  m<-length(y)
  f1<-function(x){
    (sum(proj(y-x)))-1
  }
  L<-min(y)-2/m
  U<-max(y)
  mu<-bisection(f1,L,U,10^{-4})
  x1<-proj(y-mu)
}
```

Test our function for $(-1, 1, 0.3)$:

```
## [1] "-----------------Projection onto a simplex---------------------"

## [1] 0.0000000 0.8500163 0.1500163
```

Our code works very well!

6. **Projected gradient descent:**

Let's code for the projected gradient descent method, where we project each iteration onto a simplex. Note that we need to maximize the objective function, so the sign of each iteration is negative.

```
proj_graddesc_const_step <- function(f,g,x0,tbar,epsilon, Print=FALSE, maxiter){
  stopifnot(min(x0)>=0) # check the initial point is feasible.
  x=x0
  iter=1
```

```
  trajectory<-matrix(0, nr=maxiter, nc=length(x0))
  trajectory[1,]<-x
  error<-100
  while (error > epsilon && iter<maxiter){
    iter=iter+1
    grad=g(x)
    xnew=projsimplex(x+tbar*grad)
    trajectory[iter,]<-xnew
    error = sqrt(sum((x-xnew)^2))
    if(Print){
      fun_val=f(x) # we don't need this unless we are printing it out
      print(paste('-------------- Iteration ', iter, '-------------'))
      print(c('f(x)=', f(x), 'error=', error))
      }
    x=xnew
    }
  return(list(x.opt=x, trajectory=trajectory[1:iter,], iter=iter-1))
}
```

Let's code for: $f = -||\mathbf{A}\boldsymbol{\lambda}||^2 + \sum_{i=1}^{m} \lambda_i ||\mathbf{a_i}||^2 = -\lambda^{\mathbf{T}}\mathbf{A^T}\mathbf{A}\lambda + \mathbf{b}^T\lambda$,$\nabla f = -2\mathbf{A^T}\mathbf{A}\lambda + \mathbf{b}$, where $\mathbf{b} = \left(||\mathbf{a_1}||^2, ..., ||\mathbf{a_m}||^2\right)^T$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2.., \lambda_m)^T$. (Both $\mathbf{b}$ and $\boldsymbol{\lambda}$ are column vectors).

```
A<-matrix(c(1,-4,-1,2,-3,5,-4,-3,5,2,2,-2),nc = 6)
f2<-function(x){
  #-sum((A%*%x)^2)+colSums(A^2)%*%x
  -x%*%t(A)%*%A%*%x+colSums(A^2)%*%x
}
g<-function(x){
  b<-colSums(A^2)
  -2*t(A)%*%A%*%x+b
}
```

Let's calculate the Lipschitz constant and check it numerically:

```
# Lipschitz constant
L<-2*eigen(t(A)%*%A)$values[1]
print(L)
```

```
## [1] 126.4853
```

```
Matrix::norm(2*t(A)%*%A,type = "2")
```

```
## [1] 126.4853
```

Choose an intial point and run our code:

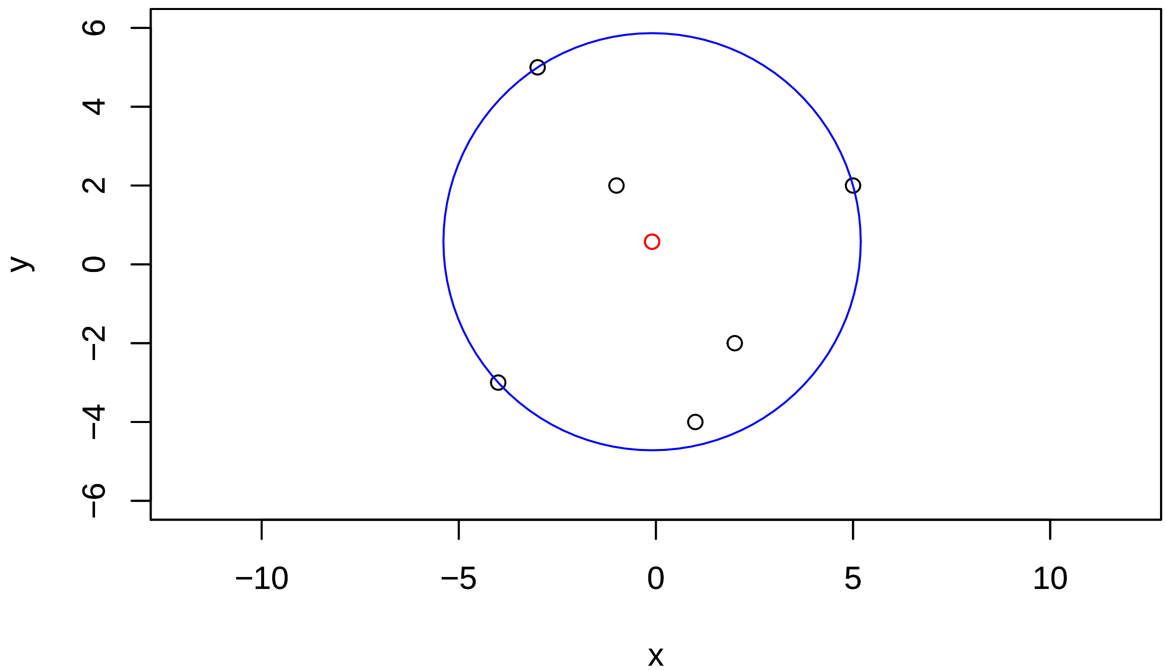
```
x0<-c(1,0,0,0,0,0)
result<-proj_graddesc_const_step(f2,g,x0,tbar=1/L,epsilon=10^{-10}, Print=FALSE, maxiter=10^4)
x_opt1<-unlist(result[1])
location<-A%*%x_opt1
```

```
## [1] "-----------------Optimal lambda--------------------"

##    x.opt1    x.opt2    x.opt3    x.opt4    x.opt5    x.opt6
## 0.0000000 0.0000000 0.1889036 0.3984070 0.4126648 0.0000000

## [1] "-----------------Optimal location x-----------------"

##             [,1]
## [1,] -0.09701493
## [2,]  0.57462687
```



Plot:

```
## [1] "-----------------Optimal radius r-----------------"

## [1] 5.292566
```

In summary: we found that there are three points on the boundary. Thus the sensor location could be built by following the above suggestions.

Use CVXR to check:

```
library(CVXR)
```

```
##
## Attaching package: 'CVXR'

## The following object is masked from 'package:stats':
##
##     power
```

```
x<-Variable(6)
b<-colSums(A^2)
objective<-Maximize(-quad_form(x,t(A)%*%A)+t(b)%*%x)
constraints<-list(x>=0,sum(x)==1)
prob<-Problem(objective,constraints)
soln<-solve(prob)
soln$status
```

```
## [1] "optimal"
```

```
x_opt2<-soln$getValue(x)
location2<-A%*%x_opt2
```

```
## [1] "-----------------Optimal lambda---------------------"
```

```
##               [,1]
## [1,] -5.929148e-23
## [2,]  9.833246e-23
## [3,]  1.889062e-01
## [4,]  3.984184e-01
## [5,]  4.126754e-01
## [6,] -4.667912e-23
```

```
## [1] "-----------------Optimal location x-----------------"
```

```
##             [,1]
## [1,] -0.09701493
## [2,]  0.57462687
```

Error analysis:

We can see the error between these two methods:

```
## [1] "-----------------Error for lambda---------------------"
```

```
##              [,1]
## [1,] 5.929148e-23
## [2,] 9.833246e-23
## [3,] 2.566031e-06
## [4,] 1.136350e-05
## [5,] 1.063038e-05
## [6,] 4.667912e-23
```

```
## [1] "-----------------Error for location x-----------------"
```

```
##              [,1]
## [1,] 2.028506e-10
## [2,] 4.137387e-10
```

Obviously, our code works very well!

Conclusion: If my company use the suggestions above, we could make our cost lowest.